

Art Unit: 2178

DETAILED ACTION

1. This office action is in response to the 1 March 2009.

This office action is made Final.

2. Claims 38-40 has been added.
3. Claims 1, 3, 6-9, 12-16, 18, 21-24, and 27-37 pending, with claims 1, 16, and 31 being the independent claims. Claims 12-15, 27-30 have been withdrawn from consideration (see below). Claims 1, 3, 6-9, 16, 18, 21-24, and 31-37 have been examined below.

Claim Rejections - 35 USC § 112

4. The following is a quotation of the first paragraph of 35 U.S.C. 112:

The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

5. Claim 38 is rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the written description requirement. The claim(s) contains subject matter which was not described in the specification in such a way as to reasonably convey to one skilled in the relevant art that the inventor(s), at the time the application was filed, had possession of the claimed invention.

Claim 38 recites the limitations “the first business logic event handler includes one or more criteria that are applied to events of the distributed document object model system and one or more business rules, and wherein the method further comprises: applying the one or more criteria to the first event; determining that the one or more criteria are satisfied by the first event; and in response to the determining, applying the first business rule that causes the third modification to the first hierarchical document.” The Applicant has stated that this new claim is supported by the specification in paragraphs 0073 and 0123. (Stated in page 16, Section V in Applicant's Remarks of the recent filing) However, the Examiner is unable to locate any disclosure in a clear, written description within the specification stating the limitations when reviewing the cited paragraphs and the rest of the specification. The Examiner is unable to locate the term “criteria” within the specification, or a clear, written description that may be viewed as “criteria”. Thus, since this feature is not described in the specification for the instant application, the examiner is forced to make a broad interpretation for this feature.

Art Unit: 2178

6. The following is a quotation of the second paragraph of 35 U.S.C. 112:

The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

7. Claim 38 is rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

8. Claim 38 recites the limitations “the first business logic event handler includes one or more criteria that are applied to events of the distributed document object model system and one or more business rules, and wherein the method further comprises: applying the one or more criteria to the first event; determining that the one or more criteria are satisfied by the first event; and in response to the determining, applying the first business rule that causes the third modification to the first hierarchical document.” However, it is unclear to the Examiner of what the limitation “criteria” means or is defined as since the claim limitation or the specification fails to give a proper clear, written definition of what criteria is. Furthermore, since this feature is not described in the specification and/or clearly defined in the claims for the instant application, the examiner is forced to make a broad interpretation for this feature.

Claim Rejections - 35 USC § 103

9. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

10. Claims 1, 3-9, 16, 18-24, 31 and 33-40 are rejected under 35 U.S.C. 103(a) as being unpatentable over Iverson, Lee, “NODAL: A File system for Ubiquitous Collaboration,” White Paper, SRI International, September 20, 2001, last downloaded by the Examiner on January 13, 2006 from <http://nodal.sourceforge.net/NODAL-WhitePaper.html>, downloaded pages 1-32

Art Unit: 2178

[hereinafter “NODAL”], in view of Iverson, Lee, “[un re-II] Meeting Summary: 4 May 2000,”

Message id: 3912508E.2CF1B4C@eng.sun.com from Erick Armstrong, May 4, 2000, last

downloaded by the Examiner on January 14, 2006, from:

<http://hot.burningchrome.com/archives/unrev-ii/msg01068.html>, downloaded pages 1-3 [hereinafter “Iverson”].

Regarding **independent claim 1**, NODAL in view of Iverson teaches:

A method in a distributed document object model system for associating business logic, comprising:

(It is noted that a “business logic” is defined in the application as including an asynchronous mode wherein a client-side business logic component may not need to wait for the DDOM client to receive a response to a mutation request before the mutation routine returns. Specifically, see Iverson (NODAL), page 27, third paragraph, stating: “Another advantage is that the server may send *any* update to the client at any time, thus fulfilling the need to notify the client when other users have modified content.”

(Emphasis in the original). The Iverson example, of notification to clients of a change in a database, is nearly identical to the embodiment described in the disclosure, as follows: “As an example, the business logic component may monitor a financial database and cause mutations to occur to a document based on changes in the database.”

The distributed document object model DDOM, is taught in Iverson, first through third paragraphs.)

receiving a first registration request from a first business logic event handler for a first event of the distributed document object model;

(See, NODAL, pages 26-27, teaching the asynchronous update routine. See also, NODAL, pages 20-21, teaching the “Cursor” interface that handles the data mutation interfaces as the business event logic

Art Unit: 2178

handler.)

registering the first business logic event handler for the first event of the distributed document object model system; the first business logic event handler is registered for a hierarchical document of the distributed document object model system,

(See, NODAL, pages 20-21, teaching that the “Cursor” is part of the software and is inherently registered to be in communication between the client and the server. Furthermore, as disclosed above, Iverson teaches a DDOM, a tree-structure form of a document. Thus, a hierarchical document is presented)

monitoring for an occurrence of at least one of the first and second events; detecting an occurrence of the first event: and in response to the occurrence of a first event, notifying the first business logic event handler; the first business logic event handler is responsive to an occurrence of the first event of the distributed document object model system

(See, NODAL, page 18-20, teaching the editing functions and the permissions for mutations are notified to the “Cursor” object.)

receiving a first indication from the first business logic event handler; and

(See, NODAL, pages 18-20, teaching messages from the “Cursor” object regarding requested mutations.)

performing a first function relating to the received first indication.

(See, NODAL, pages 18-20, teaching the editing functions.)

(NODAL teaches the business logic handler and its registration and function as claimed, but it does not expressly teach the distributed document object model (DDOM).

Iverson expressly teaches the DDOM.

It would have been obvious to one of ordinary skill in the art at the time of the invention to combine the teachings of Iverson and NODAL. Both NODAL and Iverson are in the same field of endeavor, multi-user hierarchical document editing and manipulation.

The suggestion or motivation to combine the references is that they are created by the same

Art Unit: 2178

person, Iverson reporting the developmental creation of Lee Iverson, and NODAL being authored by Lee Iverson. In addition, see, NODAL, page 11, teaching that NODAL was designed to work with a wide variety of distributed networks.)

NODAL teaches application of a business rule to cause modifications (i.e. second and/or third modifications, etc.) to a document, and the system causing the event to occur when a first modification is made (See, NODAL, at least page 20, teaching that the “Cursor” object enforces business rules related to permissions to make mutations.). See also NODAL, page 20, teaching that a program accessing the NODAL repository for any purpose other than browsing will deal primarily with the “Cursor” interfaces. See also, NODAL, page 27, teaching messaging from the server, “Cursor” object, regarding mutations.

Furthermore, it is implicitly known if the NODAL et al’s method is capable of performing the functionality once, then it may generate the same functionality over again. Thus, NODAL discloses functionality of a second business logic event handler for a second event, wherein there both are separate and different since a second handler is for a second event while a first handler is for a first event.

Regarding **dependent claim 3**, Claim 3 recites similar limitations as in Claim 1 and is similarly rejected under similar rationale. Furthermore, NODAL in view of Iverson teaches:

The method of claim 1 wherein the first business logic event handler is registered for a first document type and the second business logic event handler is registered for a second document type.

(See, NODAL, page 29, teaching the system, and inherently the “Cursor” object, enabled for an “image” type document.)

Regarding **dependent claim 6**, NODAL in view of Iverson teaches:

The method of claim 2 wherein event handling is performed on a client computing device.

Art Unit: 2178

(See, NODAL, pages 9-10, teaching that the NODAL system may be either client-side or server-side.)

Regarding **dependent claim 7**, NODAL in view of Iverson teaches:

The method of claim 2 wherein event handling is performed on a server computing device.

(See, NODAL, pages 9-10, teaching that the NODAL system may be either client-side or server-side.)

Regarding **dependent claim 8**, NODAL in view of Iverson teaches:

The method of claim 1 wherein the first event handler handles a first event that is occurs before the first modification is made to the first hierarchical document.

(See, NODAL, page 20, teaching that the “Cursor” object processes the entire content of the reference, including the mutation.)

Regarding **dependent claim 9**, NODAL in view of Iverson teaches:

The method of claim 1 wherein the first event handler handles a first event that is occurs after the first modification is made to the first hierarchical document.

(See, NODAL, page 20, teaching that the “Cursor” object maintains an audit trail after mutations are made.)

Regarding **claims 16, 18, 21-24**, claims 16, 18, 21-24 incorporate substantially similar subject matter as claimed in claims 1, 3-9, respectively, and are rejected along the same rationale.

Regarding **independent claim 31**, Claim 31 incorporates substantially similar subject matter as in claim 1, and is rejected along the same rationale. Furthermore, NODAL and Iverson fail to disclose the

Art Unit: 2178

functionality of a first event occurring again in view of a second event caused by the first action.

However, it was well-known in the art at the time of Applicant's invention that the user can perform the same modification again of the first mutation causing the functionality of a first event to occur again for the second event. The suggestion or motivation to combine the references is it enables a user to fix a mistake the second time by correcting the mistake after making the mistake originally during the first time.

Regarding **claims 32-37**, claims 32-37 incorporate substantially similar subject matter as claims in claims 1, 16, and 31 are rejected along the same rationale. Furthermore, NODAL discloses the editing of nodes. (Pages 18-20)

Regarding **claim 38**, based on the rejection of claim 38 under 35 USC 112, first and second paragraph, and the rationale incorporated, claim 38 incorporates substantially similar subject matter as claims in claims 1, 16, and 31 are rejected along the same rationale. Furthermore, pages 18-21 NODAL teaches the "cursor" object, which is the portal for accessing mutation interfaces and evaluate permissions and security of access. Also, NODAL, pages 17-18, teaching that the permissions are contained in the "user" object, which identifies the user, among other functions, and certain privileges. Thus, NODAL is able perform access of nodes based on the permissions of the privileges.

Regarding **claims 39-40**, claims 39-40 incorporate substantially similar subject matter as claims in claims 1, 16, and 31 are rejected along the same rationale.

Response to Arguments

11. Applicant's arguments filed 4 September 2009 have been fully considered but they are not persuasive.

Art Unit: 2178

On pages 13-15, in regards to Claim 1, Applicant argues that the cited references, primarily NODAL, that a Cursor is not a business logic event handler, NODAL doesn't disclose a more than one business logic event handlers, or NODAL does not teach or suggest monitoring for a modification to a hierarchical document, detecting the modification, notifying a business logic event handler, and applying a business rule that results in performing another, different, modification to the hierarchical document. However, the Examiner disagrees.

a) The claim language itself fails to disclose what a business logic handler is or how a handler is registered. In addition, the arguments presented by the Applicant fail to disclose what a business logic event handler is or how the handler is registered involve any supporting evidence from the specification stating or describing the limitation.

Therefore, from the specification defining a "business logic", it is noted that a "business logic" is defined in the application as including an asynchronous mode wherein a client-side business logic component may not need to wait for the DDOM client to receive a response to a mutation request before the mutation routine returns.(i.e. Paragraph 0126). Thus, specifically, see Iverson, page 27, third paragraph, stating: "Another advantage is that the server may send *any* update to the client at any time, thus fulfilling the need to notify the client when other users have modified content." (Emphasis in the original). The Iverson example, of notification to clients of a change in a database, is nearly identical to the embodiment described in the disclosure, as follows: "As an example, the business logic component may monitor a financial database and cause mutations to occur to a document based on changes in the database.". NODAL, pages 26-27, teaches the asynchronous update routine. NODAL, pages 20-21, teaches the "Cursor" interface that handles the data mutation interfaces as the business event logic handler. NODAL, page 27, teaches a business logic event handler such that mutations to the file may be automatically messages to a client, or may be delayed for processing. NODAL, pages 20-21, teaches that the "Cursor" is part of the software and is inherently registered to be in communication between the client and the server. NODAL teaches the business logic handler and its registration and function as claimed

b) Furthermore, the claim language itself fails to disclose occurrences of events/ the modifications to a document are monitored and/or detected. The arguments presented by the Applicant fail to disclose what these occurrences of events/modifications to a document, and/or how occurrences of events/modifications are monitored

Art Unit: 2178

by involving any supporting evidence from the specification stating or describing the limitation, or how the cited art is specifically different from Applicant's invention.

NODAL discloses on pages 18-20 of creating and editing content. NODAL discloses Editor interfaces that encapsulate all of the data mutation operations of a given node. Thus, NODAL discloses the ability to modify a node/content. Furthermore, NODAL discloses the use of a Cursor, in which, a cursor not only a reference a particular node in a repository but the entire context of that reference, including the document through which the node was accessed. Thus, the Cursor discloses the awareness of when the node was accessed such as when the node was edited (modified). In addition, NODAL on page 20 teaches the permissions for mutations are notified to the "Cursor" object, and on pages 18-20 teaches messages from the "Cursor" object regarding requested mutations. Therefore, NODAL teaches the ability of monitoring/detecting for occurrence of events and notifying.

In addition, NODAL teaches application of a business rule to cause modifications (i.e. second and/or third modifications, etc.) to a document, and the system causing the event to occur when a first modification is made (See, NODAL, at least page 20, teaching that the "Cursor" object enforces business rules related to permissions to make mutations.). See also NODAL, page 20, teaching that a program accessing the NODAL repository for any purpose other than browsing will deal primarily with the "Cursor" interfaces. See also, NODAL, page 27, teaching messaging from the server, "Cursor" object, regarding mutations.

c) Furthermore, it is implicitly known if the NODAL et al's method is capable of performing the functionality once, then it may generate the same functionality over again. Thus, NODAL discloses functionality of a second business logic event handler for a second event, wherein there both are separate and different since a second handler is for a second event while a first handler is for a first event.

Conclusion

THIS ACTION IS MADE FINAL. Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final

Art Unit: 2178

action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to David Faber whose telephone number is 571-272-2751. The examiner can normally be reached Monday-Thursday, and every other Friday.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Stephen Hong, can be reached on 571-272-4124. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

/David Faber/
Examiner, Art Unit 2178

/William L. Bashore/
Supervisory Patent Examiner, Art Unit 2175